

LAB 5: Debugging gdb

John Dempsey

COMP-232 Programming Languages
California State University, Channel Islands

February 19, 2025

Hard Due Date: February 26, 2025

Debugging gdb shows some of the basic gdb command which you can use to step through your code line-by-line, print out variables, and locate where segmentation violation errors occur.

```
john@oho:~$ mkdir LAB5
```

```
john@oho:~$ cd LAB5
```

```
john@oho:~/LAB5$ sftp john@comp232.com
```

```
john@comp232.com's password:
```

```
Connected to comp232.com.
```

```
sftp> cd /home/LAB5
```

```
sftp> ls -l
```

```
-rw-r--r-- 1 john john 334 Feb 21 05:20 Makefile
-rw-r--r-- 1 john john 211 Feb 21 05:20 getComputerMove.c
-rw-r--r-- 1 john john 308 Feb 21 05:20 getHumanMove.c
-rw-r--r-- 1 john john 101 Feb 21 05:20 initBoard.c
-rw-r--r-- 1 john john 843 Feb 21 05:20 is3inARow.c
-rw-r--r-- 1 john john 291 Feb 21 05:20 isAWin.c
-rw-r--r-- 1 john john 248 Feb 21 05:20 isBoardFull.c
-rw-r--r-- 1 john john 86 Feb 21 05:20 main.c
-rw-r--r-- 1 john john 156 Feb 21 05:20 makeMove.c
-rw-r--r-- 1 john john 374 Feb 21 05:20 max.c
-rw-r--r-- 1 john john 374 Feb 21 05:20 min.c
-rw-r--r-- 1 john john 919 Feb 21 05:20 minMax.c
-rw-r--r-- 1 john john 280 Feb 21 05:20 printBoard.c
-rw-r--r-- 1 john john 1327 Feb 21 05:20 runGame.c
-rw-r--r-- 1 john john 607 Feb 21 05:20 tictactoe.h
```

```
sftp> mget *
```

```
Fetching /home/LAB5/Makefile to Makefile
/home/LAB5/Makefile 100% 334
5.0KB/s 00:00
Fetching /home/LAB5/getComputerMove.c to getComputerMove.c
/home/LAB5/getComputerMove.c 100% 211
3.6KB/s 00:00
Fetching /home/LAB5/getHumanMove.c to getHumanMove.c
```

```

/home/LAB5/getHumanMove.c                                100%  308
5.9KB/s   00:00
Fetching /home/LAB5/initBoard.c to initBoard.c
/home/LAB5/initBoard.c                                  100%  101
1.8KB/s   00:00
Fetching /home/LAB5/is3inARow.c to is3inARow.c
/home/LAB5/is3inARow.c                                  100%  843
12.5KB/s  00:00
Fetching /home/LAB5/isAWin.c to isAWin.c
/home/LAB5/isAWin.c                                    100%  291
3.8KB/s   00:00
Fetching /home/LAB5/isBoardFull.c to isBoardFull.c
/home/LAB5/isBoardFull.c                                100%  248
4.2KB/s   00:00
Fetching /home/LAB5/main.c to main.c
/home/LAB5/main.c                                      100%   86
1.5KB/s   00:00
Fetching /home/LAB5/makeMove.c to makeMove.c
/home/LAB5/makeMove.c                                  100%  156
2.4KB/s   00:00
Fetching /home/LAB5/max.c to max.c
/home/LAB5/max.c                                        100%  374
6.7KB/s   00:00
Fetching /home/LAB5/min.c to min.c
/home/LAB5/min.c                                        100%  374
5.8KB/s   00:00
Fetching /home/LAB5/minMax.c to minMax.c
/home/LAB5/minMax.c                                    100%  919
13.8KB/s  00:00
Fetching /home/LAB5/printBoard.c to printBoard.c
/home/LAB5/printBoard.c                                100%  280
4.7KB/s   00:00
Fetching /home/LAB5/runGame.c to runGame.c
/home/LAB5/runGame.c                                  100% 1327
23.3KB/s  00:00
Fetching /home/LAB5/tictactoe.h to tictactoe.h
/home/LAB5/tictactoe.h                                100%  607
10.6KB/s  00:00

```

sftp> quit

john@oho:~/LAB5\$ ls -l

total 16

```

-rw-r--r-- 1 john john 334 Feb 20 21:27 Makefile
-rw-r--r-- 1 john john 211 Feb 20 21:27 getComputerMove.c
-rw-r--r-- 1 john john 308 Feb 20 21:27 getHumanMove.c
-rw-r--r-- 1 john john 101 Feb 20 21:27 initBoard.c
-rw-r--r-- 1 john john 843 Feb 20 21:27 is3inARow.c
-rw-r--r-- 1 john john 291 Feb 20 21:27 isAWin.c
-rw-r--r-- 1 john john 248 Feb 20 21:27 isBoardFull.c
-rw-r--r-- 1 john john 86 Feb 20 21:27 main.c
-rw-r--r-- 1 john john 156 Feb 20 21:27 makeMove.c
-rw-r--r-- 1 john john 374 Feb 20 21:27 max.c

```

```
-rw-r--r-- 1 john john 374 Feb 20 21:27 min.c
-rw-r--r-- 1 john john 919 Feb 20 21:27 minMax.c
-rw-r--r-- 1 john john 280 Feb 20 21:27 printBoard.c
-rw-r--r-- 1 john john 1327 Feb 20 21:27 runGame.c
-rw-r--r-- 1 john john 607 Feb 20 21:27 tictactoe.h
```

```
john@oho:~/LAB5$ cat Makefile
# Makefile
```

```
SOURCE=\
    getComputerMove.c\  
    getHumanMove.c\  
    initBoard.c\  
    is3inARow.c\  
    isAWin.c\  
    isBoardFull.c\  
    makeMove.c\  
    max.c\  
    min.c\  
    minMax.c\  
    printBoard.c\  
    runGame.c
```

```
CFLAGS = -m64 -g -w
```

```
OBJS = $(SOURCE:.c=.o)
```

```
tictactoe: $(OBJS) main.o
    gcc main.c $(CFLAGS) $(OBJS) -lm -o $@
    ls -l tictactoe
```

```
clean:
    rm *.o tictactoe
```

```
john@oho:~/LAB5$ make
cc -m64 -g -w -c -o getComputerMove.o getComputerMove.c
cc -m64 -g -w -c -o getHumanMove.o getHumanMove.c
cc -m64 -g -w -c -o initBoard.o initBoard.c
cc -m64 -g -w -c -o is3inARow.o is3inARow.c
cc -m64 -g -w -c -o isAWin.o isAWin.c
cc -m64 -g -w -c -o isBoardFull.o isBoardFull.c
cc -m64 -g -w -c -o makeMove.o makeMove.c
cc -m64 -g -w -c -o max.o max.c
cc -m64 -g -w -c -o min.o min.c
```

```
cc -m64 -g -w -c -o minMax.o minMax.c
cc -m64 -g -w -c -o printBoard.o printBoard.c
cc -m64 -g -w -c -o runGame.o runGame.c
cc -m64 -g -w -c -o main.o main.c
gcc main.c -m64 -g -w getComputerMove.o getHumanMove.o initBoard.o is3inARow.o isAWin.o
isBoardFull.o makeMove.o max.o min.o minMax.o printBoard.o runGame.o -lm -o tictactoe
ls -l tictactoe
-rwxr-xr-x 1 john john 39416 Feb 20 21:27 tictactoe
```

```
john@oho:~/LAB5$ tictactoe
```

```
^C
```

```
john@oho:~/LAB5$ gdb tictactoe
```

```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.
```

```
For help, type "help".
```

```
Type "apropos word" to search for commands related to "word"...
```

```
Reading symbols from tictactoe...
```

```
(gdb) run
```

← Type run in the gdb debugger.

```
Starting program: /home/john/LAB5/tictactoe
```

```
^C
```

← Nothing happening. Type ^C to interrupt program.

```
Program received signal SIGINT, Interrupt.
```

```
runGame () at runGame.c:18
```

```
18         j++;
```

```
(gdb) bt
#0 runGame () at runGame.c:18
#1 0x00000000080011e6 in main (argc=1, argv=0x7ffffffdfb8) at main.c:5
```

(gdb) n ← Type n for next line.

(gdb) n ← Type n for next line.

(gdb) n ← Type n for next line.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

Looks like we're in an infinite loop.

(gdb) l ← Type lower case L to list source code.

```
9      int k = 0;
10
11     // ERROR: Infinite loop. Comment out entire while loop.
12     // Use "watch i" and "watch j" to see values change.
13     while (i<100) {
14         if (i < 50) {
15             i++;
16         }
17         else {
18             j++;
```

Let's get a better view of what our program is doing.

(gdb) layout src

(gdb) n ← Type n for next line.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

(gdb) <CR> ← Hit Carriage Return to repeat previous command.

It's a loop alright.

(gdb) **layout split** ← Displays source code and assembly language code being executed.

(gdb) **n** ← Type **n** for next line.

(gdb) **<CR>** ← Hit Carriage Return to repeat previous command.

(gdb) **<CR>** ← Hit Carriage Return to repeat previous command.

(gdb) **<CR>** ← Hit Carriage Return to repeat previous command.

(gdb) **layout src**

(gdb) **frame** ← Let's see where we are in the code.

#0 runGame () at runGame.c:18

18 j++;

(gdb) **print j** ← Print the j value. Whoa! That's really a large number.

\$1 = 1622339873

(gdb) **n**

13 while(i<100) {

(gdb) **n**

14 if (i < 50) {

(gdb) **p i** ← Print the i value. P is short for print.

\$2 = 50

(gdb) **n**

(gdb) **p j**

\$3 = 1622339874

(gdb) **n**

13 while(i<100) {

(gdb) **p j**

\$4 = 1622339875

(gdb) **watch i**

Watchpoint 2: i

```
(gdb) watch j
Watchpoint 3: j
```

```
(gdb) n ← Run the next line of code. See the j value change using watch.
```

```
(gdb) n
```

```
(gdb) n
```

Ok. Let's edit the code to remove the infinite loop. You can comment out the entire while loop.

```
(gdb) !!s -l runGame.c
-rw-r--r-- 1 john john 1327 Feb 20 21:27 runGame.c
```

```
(gdb)!vi runGame.c
```

```
(gdb) quit
A debugging session is active.
```

```
Inferior 1 [process 1453156] will be killed.
```

```
Quit anyway? (y or n) y
```

After editing the runGame.c function to remove the infinite loop, you will need to recompile the programs by running "make" again and rerunning the tictactoe program.

Use the gdb cheat sheet to learn more commands.

Upload gdb.txt File to Receive Credit to comp232.com

To receive credit for **Debugging GDB**, create a text file called gdb.txt and copy/paste the tic-tac-toe output that results in a cat game or you win game.

SFTP the gdb.txt file to your LAB5 directory on comp232.com.